

[MS-OXONOTE]: Note Object Protocol Specification

Intellectual Property Rights Notice for Protocol Documentation

- **Copyrights.** This protocol documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the protocols, and may distribute portions of it in your implementations of the protocols or your documentation as necessary to properly document the implementation. This permission also applies to any documents that are referenced in the protocol documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the protocols. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, the protocols may be covered by Microsoft's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp/default.mspx>). If you would prefer a written license, or if the protocols are not covered by the OSP, patent licenses are available by contacting protocol@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Preliminary Documentation. This documentation is preliminary documentation for these protocols. Since the documentation may change between this preliminary version and the final version, there are risks in relying on preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Tools. This protocol documentation is intended for use in conjunction with publicly available standard specifications and networking programming art, and assumes that the reader is either familiar with the aforementioned material or has immediate access to it. A protocol specification does not require the use of Microsoft programming tools or programming environments in order for a Licensee to develop an implementation. Licensees who have access to Microsoft programming tools and environments are free to take advantage of them.

Revision Summary			
Author	Date	Version	Comments
Microsoft Corporation	April 4, 2008	0.1	Initial Availability

Preliminary

Table of Contents

1	<i>Introduction</i>	4
1.1	Glossary.....	4
1.2	References.....	4
1.2.1	Normative References.....	4
1.2.2	Informative References.....	5
1.3	Protocol Overview (Synopsis).....	5
1.4	Relationship to Other Protocols.....	5
1.5	Prerequisites/Preconditions.....	5
1.6	Applicability Statement.....	6
1.7	Versioning and Capability Negotiation.....	6
1.8	Vendor-Extensible Fields.....	6
1.9	Standards Assignments.....	6
2	<i>Messages</i>	6
2.1	Transport.....	6
2.2	Message Syntax.....	6
2.2.1	Note Object Properties.....	6
2.2.2	Additional Property Constraints.....	7
3	<i>Protocol Details</i>	8
3.1	Common Details.....	8
3.1.1	Abstract Data Model.....	8
3.1.2	Timers.....	9
3.1.3	Initialization.....	9
3.1.4	Higher-Layer Triggered Events.....	9
3.1.5	Message Processing Events and Sequencing Rules.....	9
3.1.6	Timer Events.....	9
3.1.7	Other Local Events.....	9
4	<i>Protocol Examples</i>	10
4.1	Sample Note Object.....	10
5	<i>Security</i>	11
5.1	Security Considerations for Implementers.....	11
5.2	Index of Security Parameters.....	11
6	<i>Appendix A: Office/Exchange Behavior</i>	11
	<i>Index</i>	13

1 Introduction

This document specifies the Note Object Protocol, which defines properties of an object that models the electronic equivalent to a "sticky note". A Note Object presents the user with a very simple interface for keeping brief notes.

1.1 Glossary

The following terms are defined in [MS-OXGLOS]:

attachment object
best body
folder object
message object
named property
plain-text message body
property
property identifier or property ID
property type
recipient
special folder
Unicode

The following terms are specific to this document:

note object: A **message object** that represents a simple text note in a messaging store and that adheres to the **property** specifications in this document. A **note object** models the electronic equivalent to a "sticky note".

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

1.2.1 Normative References

[MS-OXBBODY] Microsoft Corporation, "Best Body Retrieval Protocol Specification", April 2008.

[MS-OXCFOLD] Microsoft Corporation, "Folder Object Protocol Specification", April 2008.

[MS-OXCMSG] Microsoft Corporation, "Message and Attachment Object Protocol Specification", April 2008.

[MS-OXCPRPT] Microsoft Corporation, "Property and Stream Object Protocol Specification", April 2008.

[MS-OXGLOS] Microsoft Corporation, "Office Exchange Protocols Master Glossary", April 2008.

[MS-OXOSFLD] Microsoft Corporation, "Special Folders Protocol Specification", April 2008.

[MS-OXPROPS] Microsoft Corporation, "Office Exchange Protocols Master Property List Specification", April 2008.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://www.ietf.org/rfc/rfc2119.txt>.

1.2.2 Informative References

[MS-OXCDATA] Microsoft Corporation, "Data Structures Protocol Specification", April 2008.

1.3 Protocol Overview (Synopsis)

The Note Object Protocol allows the representation of a simple text note in a messaging store. The Note Object Protocol extends the Message and Attachment Object Protocol in that it defines new properties and adds restrictions to the properties that are defined in [MS-OXCMSG].

A **note object** represents a "sticky note." The properties that are specific to a **note object** facilitate retaining information about the note's background color, window location, and size. A **note object** contains simple text (that is, text with little formatting other than line breaks), and is stored in a **folder object**. The Note Object Protocol also specifies how a **note object** is created and manipulated.

1.4 Relationship to Other Protocols

The Note Object Protocol has the same dependencies as the Message and Attachment Object Protocol, which it extends. For details about the Message and Attachment Object Protocol, see [MS-OXCMSG].

1.5 Prerequisites/Preconditions

The Note Object Protocol has the same prerequisites and preconditions as the Message and Attachment Object Protocol.

1.6 Applicability Statement

None.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

This protocol provides no vendor-extensibility beyond what is already specified in [MS-OXCMSG].

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The Note Object protocol uses the protocols defined in [MS-OXCPRPT] and [MS-OXCMSG] as its primary transport mechanism.

2.2 Message Syntax

A **note object** can be created and modified by clients and servers. Except where noted below, this section defines constraints under which both clients and servers operate.

Clients operate on **note objects** using the Message and Attachment Object Protocol, as specified in [MS-OXCMSG]. How a server operates on **note objects** is implementation-dependent. The results of any such operation are exposed to clients in a manner that is consistent with the Note Object Protocol.

Unless otherwise specified below, a **note object** adheres to all property constraints specified in [MS-OXPROPS] and all property constraints specified in [MS-OXCMSG]. A note object MAY also contain other properties <1> <2>, which are defined in [MS-OXPROPS], but these properties have no impact on the Note Object Protocol.

2.2.1 Note Object Properties

The following properties are specific to **note objects**.

2.2.1.1 PidLidNoteColor

Type: PtyInteger32.

Specifies the suggested background color of the **note object**; MUST be one of the entries in the following table <3>.

Value	Color
0x00000000	Blue
0x00000001	Green
0x00000002	Pink
0x00000003	Yellow
0x00000004	White

2.2.1.2 PidLidNoteWidth

Type: PtypInteger32, signed.

Specifies the width of the visible message window in pixels; the value **MUST** be greater than zero.

2.2.1.3 PidLidNoteHeight

Type: PtypInteger32, signed.

Specifies the height of the visible message window in pixels; the value **MUST** be greater than zero.

2.2.1.4 PidLidNoteX

Type: PtypInteger32, signed.

Specifies the distance, in pixels, from the left edge of the screen that a user interface displays a **note object**.

2.2.1.5 PidLidNoteY

Type: PtypInteger32, signed.

Specifies the distance, in pixels, from the top edge of the screen that a user interface displays a **note object**.

2.2.2 Additional Property Constraints

This protocol specifies additional constraints on the following properties beyond what is specified in [MS-OXCMSG].

2.2.2.1 Best Body Properties

The contents of the **note object**. A **plain-text message body** stored as specified in [MS-OXCMSG]<4>.

2.2.2.2 PidTagIconIndex

Type: PtypInteger32.

Specifies which icon is to be used by a user interface when displaying a group of **note objects**. The value MUST be 0x00000300 added to the value of PidLidNoteColor.

2.2.2.3 PidTagMessageClass

Type: PtypString8, case-insensitive.

Specifies the type of the **message item**. The value MUST be “IPM.StickyNote” or begin with “IPM.StickyNote.”, in addition to meeting the criteria specified in [MS-OXCMSG].

2.2.2.4 PidTagNormalizedSubject

Type: PtypString.

Specifies an abbreviated version of the contents of the note suitable for displaying groups of **note objects** to a user.

2.2.2.5 Recipients

A note object MUST NOT have **recipients**.

2.2.2.6 Attachment Objects

A **note object** MUST NOT have **attachment objects**.

3 Protocol Details

General protocol details, as specified in [MS-OXPROPS] and [MS-OXCMSG], apply.

3.1 Common Details

The client and server roles are to create and manipulate electronic sticky notes, and otherwise operate in their roles as specified in [MS-OXCMSG].

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

3.1.1.1 Note Objects

A **note object** extends the **message object** as defined in [MS-OXCMSG].

3.1.1.2 Note Special Folder

A **note object** is created in the Notes **special folder** as defined in [MS-OXOSFLD] unless the **end-user** or **user agent** explicitly specifies another **folder object**.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Creation of a Note Object

To create a **note object**, the server or client creates a message object as specified in [MS-OXCMSG], sets properties in accordance with the requirements in section 2 and [MS-OXCPRPT], and saves the resulting **message object** as specified in [MS-OXCMSG].

3.1.4.2 Modification of a Note Object

When modifying a **note object**, the client or server opens a message object as specified in [MS-OXCMSG], modifies any of the properties in accordance with the requirements in section 2 and [MS-OXCPRPT], and saves the message object as specified in [MS-OXCMSG].

3.1.4.3 Deletion of a Note Object

Note objects have no special semantics in relation to deletion beyond what is defined in [MS-OXCFOLD].

3.1.5 Message Processing Events and Sequencing Rules

None.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

4 Protocol Examples

4.1 Sample Note Object

Joe creates a **note object**, types in his grocery list, and saves it. The following is a description of what a client might do to accomplish Joe's intentions and the responses a server might return. See [MS-OXCPRPT] and [MS-OXCMSG] for details on ROPs.

Before manipulating **note objects**, the client needs to ask the server to perform a mapping from **named properties** to **property identifiers**, using RopGetPropertyIdsFromNames:

Property	Property Set GUID	NameID
PidLidNoteColor	{0006200E-0000-0000-C000-000000000046}	0x00008B00
PidLidNoteWidth	{0006200E-0000-0000-C000-000000000046}	0x00008B02
PidLidNoteHeight	{0006200E-0000-0000-C000-000000000046}	0x00008B03
PidLidNoteX	{0006200E-0000-0000-C000-000000000046}	0x00008B04
PidLidNoteY	{0006200E-0000-0000-C000-000000000046}	0x00008B05

The server might respond with the following identifiers, which will be used in the example that follows. (The actual identifiers are at the discretion of the server.)

Property	Property ID
PidLidNoteColor	0x8046
PidLidNoteWidth	0x8047
PidLidNoteHeight	0x8048
PidLidNoteX	0x8049
PidLidNoteY	0x804A

To create a note object, the client uses RopCreateMessage. The server returns a success code and a handle to a message object.

After Joe has input his content for the note object, the client uses RopSetProperties to transmit his data to the server. See [MS-OXCDATA] for details on property types.

Property	Property ID	Data Type	Value
PidLidNoteColor	0x8046	0x0003 (PtypInteger32)	0x00000003
PidLidNoteWidth	0x8047	0x0003 (PtypInteger32)	0x000000C8
PidLidNoteHeight	0x8048	0x0003 (PtypInteger32)	0x000000A6
PidLidNoteX	0x8049	0x0003 (PtypInteger32)	0x0000006E
PidLidNoteY	0x804A	0x0003 (PtypInteger32)	0x0000006E
PidTagIconIndex	0x1080	0x0003 (PtypInteger32)	0x00000303
PidTagMessageClass	0x001A	0x001E (PtypString8)	IPM.StickyNote
PidTagNormalizedSubject	0x0e1d	0x001f (PtypString)	Grocery List
PidTagSubjectPrefix	0x003d	0x001f (PtypString)	(null)
PidTagBody	0x1000	0x001f (PtypString)	"Grocery List:"

Property	Property ID	Data Type	Value
			Celery Broccoli"

When Joe is ready to save his changes, the client uses RopSaveChangesMessage to commit the properties on the server, and then RopRelease to release the **note object**.

The values of some properties will change during the execution of RopSaveChangesMessage, but the properties specified in this document will not change.

5 Security

5.1 Security Considerations for Implementers

There are no special security considerations specific to the Note Object protocol. General security considerations pertaining to the underlying transport apply, as specified in [MS-OXCMSG] and [MS-OXCPRPT].

5.2 Index of Security Parameters

None.

6 Appendix A: Office/Exchange Behavior

The information in this specification is applicable to the following versions of Office/Exchange:

- Microsoft Office 2003 with Service Pack 3 applied
- Microsoft Exchange 2003 with Service Pack 2 applied
- Microsoft Office 2007 with Service Pack 1 applied
- Microsoft Exchange 2007 with Service Pack 1 applied

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies Office/Exchange behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies Office/Exchange does not follow the prescription.

<1> Section 2.2.1: "Microsoft Office Outlook 2003" and "Microsoft Office Outlook 2007" sometimes set the following properties regardless of user input; their values have no meaning in the context of this protocol.

PidLidAgingDontAgeMe, PidLidCurrentVersion, PidLidCurrentVersionName, PidLidPrivate, PidLidSideEffect, PidTagAlternateRecipientAllowed,

PidTagClientSubmitTime, PidTagDeleteAfterSubmit, PidTagImportance,
PidTagMessageDeliveryTime, PidTagPriority, PidTagReadReceiptRequested,
PidTagSensitivity, PidLidReminderDelta, PidLidReminderSet, PidLidReminderNextTime,
PidLidTaskMode

<2> Section 2.2.1: “Microsoft Office Outlook 2007” sets the following properties regardless of user input; their values have no meaning in the context of this protocol.

PidLidPercentComplete, PidLidTaskActualEffort, PidLidTaskComplete,
PidLidTaskAssigner, PidLidTaskAcceptanceState, PidLidTaskEstimatedEffort,
PidLidTaskFFixOffline, PidLidTaskFRecurring, PidLidTaskNoCompute, PidLidTaskOrdinal,
PidLidTaskOwnership, PidLidTaskRole, PidLidTaskState, PidLidTaskStatus,
PidLidTaskVersion, PidLidTeamTask, PidLidValidFlagStringProof

<3> “Microsoft Office Outlook 2003” will always use PidLidNoteColor to determine the background color, regardless of the existence or value of PidNameKeywords. “Microsoft Office Outlook 2007” ignores PidLidNoteColor if the item has PidNameKeywords set also – in that case, the background color will be the color associated with the first keyword listed, as specified in [MS-OXOCFG].

<4> “Microsoft Office Outlook 2003” and “Microsoft Office Outlook 2007” set encapsulated plain text as a Rich Text Body as specified in [MS-OXRTFEX] and [MS-OXCMSG].

Index

- Introduction, 4
 - Applicability statement, 6
 - Glossary, 4
 - Prerequisites/Preconditions, 5
 - Protocol overview (synopsis), 5
 - References, 4
 - Relationship to other protocols, 5
 - Standards assignments, 6
 - Vendor-extensible fields, 6
 - Versioning, 6
- Messages, 6
 - Message syntax, 6
 - Transport, 6
- Office/Exchange behavior, 11
- Protocol details, 8
 - Common details, 8
- Protocol examples, 10
 - Sample note object, 10
- References
 - Informative references, 5
 - Normative references, 4
- Security, 11
 - Index of security parameters, 11
 - Security considerations for implementers, 11